GRAFICA

C. 1985 by
ARTI GRAFICHE RICORDI
Prodotto dalla Enda S.r.l.

VOLUME 2

Affrontiamo in questo secondo volume **IL MOVIMENTO**: cioè come spostare, sovrapporre, far camminare, ruotare o scontrare più sprites o caratteri.
Un procedimento utilissimo non solo perchè applicabile ai giochi, ma anche per l'uso didattico che ne deriva: ad esempio in campo geometrico, matematico, scientifico e nei programmi gestionali in genere.
Naturalmente gli esempi che arricchiscono il programma, oltre a poter essere listati direttamente sul video, sono riportati nelle pagine che seguono, così da poter essere studiati con comodità.

Priorità sprites CBM 64

```
1000 rem *** priorita' sprite ***
1010 rem
1020 v=53248
1025 forKK=0to15:poKev+KK,0:nextKK
1030 poKev+23,255:poKev+29,255
1035 forx=0to7
1040 poKev+39+x,x+7
1045 nextx
1050 forx=0to62
1055 reada
1060 poKe12288+x,a
1065 nextx
1070 data 0,24,0,0,60,0,0,126,0,0,255,0
1075 data 1,255,128,3,255,192,7,255,224
1080 data 15,255,240,31,255,248,63,255,252
1085 data 127,255,254,63,255,252,31,255
1090 data 248,15,255,240,7,255,224,3,255
1095 data 192,1,255,128,0,255,0,0,126,0
1100 data 0,60,0,0,24,0
1105 poKe2040,192:poKe2041,192:poKe2042,192
1110 poKe2043,192:poKe2044,192:poKe2045,192
1115 poKe2046,192:poKe2047,192
1120 poKev+21,255
1125 forx=0to14step2
1130 poKev+x,100
1135 nextx
1140 fory=1to15step2
1145 poKev+y,80
1150 nexty
1155 forx=0to14step2
1160 forz=0to100
1165 poKev+x,100+z
1170 nextz
1175 nextx
1180 fory=1to15step2
1185 forz=0to100
1190 poKev+y,80+z
1195 nextz
1200 nexty
1205 forx=0to14step2
1210 forz=1to10*x
1215 poKev+x,200-z
1220 nextz
1225 nextx
1230 fory=1to15step2
1235 forz=1to5*y
1240 poKev+y,180-z
```

```
1245 nextz
1250 nexty
1255 z=8
1260 forKK=1to50
1265 poKev+39+z,15*rnd(0)
1270 z=z-1:ifz=0thenz=8
1275 ford=1to50:nextd
1280 nextKK:poKev+21,0
```

Collisioni sprites CBM 64

```
2000 rem *** collisioni sprite ***
2010 rem
2020 v=53248
2030 forKK=0to3:poKev+KK,0:nextKK
2040 poKev+23,3:poKev+29,3
2050 poKev+39,7:poKev+40,8
2060 poKe2040,192:poKe2041,192
2070 poKev,120:poKev+2,200
2080 poKev+1,100:poKev+3,100
2090 forx=0to62
2100 reada:poKe12288+x,a
2110 nextx
2120 poKev+21,3
2130 forKK=1to3:forx=0to150
2140 poKev+2,200-x
2150 print"██LOCAZIONE COLLISIONI (V+30):";peeK(v+30)"█    "
2160 nextx
2170 nextKK:poKev+21,0
2200 data 0,24,0,0,60,0,0,126,0,0,255,0
2210 data 1,255,128,3,255,192,7,255,224
2220 data 15,255,240,31,255,248,63,255,252
2230 data 127,255,254,63,255,252,31,255
2240 data 248,15,255,240,7,255,224,3,255
2250 data 192,1,255,128,0,255,0,0,126,0
2260 data 0,60,0,0,24,0
```

Rotazione CBM 64

```
3018 rem *** rotazione ***
3020 jj=0:v=53248
3030 pokev+39,50
3040 pokev,0:pokev+1,0
3050 pokev+1,200
3060 forx=0to62
3070 reada
3080 poke 12288+x,a
3090 next x
3100 for x=0to62
3110 reada
3120 poke 12352+x,a
3130 next x
3140 for x=0to62
3150 reada
3160 poke 12416+x,a
3170 next x
3180 pokev+21,1
3190 data 255,255,255,255,0,1,255,0,1
3200 data 255,0,1,255,0,1,255,0,1
3210 data 255,0,1,255,0,1,255,0,1
3220 data 255,0,1,255,0,1,255,0,1
3230 data 255,0,1,255,0,1,255,0,1
3240 data 255,0,1,255,0,1,255,0,1
3250 data 255,0,1,255,0,1,255,255,255
3260 data 255,255,255,128,255,1,128,255,1
3270 data 128,255,1,128,255,1,128,255,1
3280 data 128,255,1,128,255,1,128,255,1
3290 data 128,255,1,128,255,1,128,255,1
3300 data 128,255,1,128,255,1,128,255,1
3310 data 128,255,1,128,255,1,128,255,1
3320 data 128,255,1,128,255,1,255,255,255
3330 data 255,255,255,128,0,255,128,0,255
3340 data 128,0,255,128,0,255,128,0,255
3350 data 128,0,255,128,0,255,128,0,255
3360 data 128,0,255,128,0,255,128,0,255
3370 data 128,0,255,128,0,255,128,0,255
3380 data 128,0,255,128,0,255,128,0,255
3390 data 128,0,255,128,0,255,255,255,255
3400 pokev+23,1:pokev+29,1
3410 for x=0 to 2
3420 poke 2040,192+x
3430 poke v,q:q=q+5:jj=jj+5:if q>250 then q=20
3440 ifjj>750thenpokev+21,0:goto3480
3450 for d=1 to 50:next d
3460 next x
3470 goto3410
```

Animale che cammina CBM 64

```
4500 rem *** animale che cammina ***
4510 rem
4520 jj=0:v=53248
4530 forKK=0to3:poKev+KK,0:nextKK
4540 poKev+23,1:poKev+29,1
4550 poKev+39,50
4560 poKev+1,200
4570 forx=0to62
4580 reada
4590 poKe12288+x,a
4600 next x
4610 forx=0to62
4620 reada
4630 poKe12352+x,a
4640 nextx
4650 data 0,0,0,0,0,0,0,0,144
4660 data 0,0,96,0,0,240,0,1,248
4670 data 131,255,252,71,255,238,47,255,196
4680 data 31,255,128,15,255,128,15,255,128
4690 data 15,255,128,20,1,64,20,1,64
4700 data 20,1,64,36,1,32,72,0,144
4710 data 144,0,72,144,0,72,0,0,0
4720 data 0,0,0,0,0,0,0,0,144
4730 data 0,0,96,0,0,240,0,1,248
4740 data 3,255,252,7,255,238,15,255,196
4750 data 255,255,128,15,255,128,15,255,128
4760 data 15,255,128,20,1,64,20,1,64
4770 data 20,1,64,20,1,64,18,2,80
4780 data 9,4,128,9,4,128,0,0,0
4790 poKe 2040,192
4800 poKev+21,1
4810 poKe v,q:q=q+3:jj=jj+3:if q>250 then q=20
4820 ifjj>300thenpoKev+21,0:goto4880
4830 for d=1 to 50:next d
4840 poKe v,q:q=q+3
4850 poKe 2040,193
4860 for d=1 to 50:next d
4870 goto 4790
```

Uomo che cammina CBM 64

```
6000 rem *** uomo che cammina ***
6010 rem
6020 jj=0:v=53248
6030 poKev,0:poKev+1,0
6040 poKe v+39,7
6050 poKe v+28,1
6060 poKe v+37,2
6070 poKe v+38,3
```

```
6080 poKe v+29,1
6090 poKe v+23,1
6100 poKe v+1,100
6110 for x=0to62
6120 reada:poKe 12288+x,a
6130 next x
6140 for x=0to62
6150 reada:poKe 12352+x,a
6160 next x
6170 rem dati per sprite 1
6180 data 0,84,0,0,84,0,0,84,0
6190 data 0,84,0,0,32,0,0,32,0
6200 data 0,168,0,0,168,0,2,34,0
6210 data 2,34,0,8,32,128,8,32,128
6220 data 0,32,0,0,48,0,0,48,0
6230 data 0,48,0,0,48,0,0,48,0
6240 data 0,48,0,0,60,0,0,60,0
6250 rem dati per sprite 2
6260 data 0,84,0,0,84,0,0,84,128
6270 data 0,84,128,0,34,0,0,34,0
6280 data 0,168,0,0,168,0,2,32,0
6290 data 2,32,0,8,32,0,8,32,0
6300 data 0,32,0,0,60,0,0,60,0
6310 data 0,51,48,0,51,48,0,48,192
6320 data 0,48,192,0,60,0,0,60,0
6330 poKe v+0,30
6340 poKe 2040,192
6350 poKev+21,1
6360 poKe v+0,q:q=q+5:jj=jj+5:if q>250 then q=30
6370 ifjj>200thenpoKev+21,0:poKev+28,0:end
6380 for d=1 to 50:next d
6390 poKe 2040,193
6400 poKe v+0,q:q=q+5
6410 rem delay
6420 for d=1 to 100:next d
6430 goto 6340
```

Movimento carattere VIC 20

```
10 printchr$(147)
20 poKe36879,120
30 v=7680:c=38400
40 forK=495to11step-22
50 poKev+K,65:poKec+K,0
60 fory=1to50:nexty
70 poKev+K,32:poKec+K,7
80 nextK
```

Collisioni di caratteri VIC 20

```
10 printchr$(147)
20 poke36879,120
30 v=7680:c=38400
35 pokev+77,66:pokec+77,6
40 forK=485to11step-22
45 p=peeK(v+K)
46 print"██PEEK("v+K")="p
47 ifp=66thenpokev+K,42:end
50 pokev+K,65:pokec+K,0
60 fory=1to100:nexty
70 pokev+K,32:pokec+K,7
80 nextk
```

Uomo che cammina VIC 20

```
10 poke52,28:poke56,28:poke51,0:poke55,0
40 printchr$(147)
1020 forK=0to63*8+7
1040 poke7168+K,peeK(32768+K)
1060 nextK
1080 forK=58*8to59*8+7
1100 reada
1120 poke7168+K,a
1140 nextK
1160 poke36869,255
1200 data24,24,60,90,24,24,20,18
1220 data25,26,60,88,24,24,40,72
2000 poke36879,120
2020 v=7680:c=38400
2040 forK=440to461
2060 pokev+K,58:pokec+K,0
2080 fory=1to100:nexty
2090 pokev+K,59
2100 fory=1to100:nexty
2120 pokev+K,32:pokec+K,7
2200 nextK
3000 print"█":poke36869,240
```

Tabella codici ASCII

CAR	COD	CAR	COD	CAR	COD	CAR	COD	CAR	COD	CAR	COD
	0	!	33	B	66	─	99		132	▏	165
	1	"	34	C	67	▔	100	f1	133	▒	166
	2	#	35	D	68	▔	101	f3	134	▕	167
	3	$	36	E	69	▁	102	f5	135	▄	168
	4	%	37	F	70	▏	103	f7	136	◤	169
WHT	5	&	38	G	71	▕	104	f2	137	▕	170
	6	'	39	H	72	╮	105	f4	138	├	171
	7	(	40	I	73	╰	106	f6	139	▗	172
disabilita SHIFT C=	8	)	41	J	74	╯	107	f8	140	└	173
abilita SHIFT C=	9	*	42	K	75	└	108	SHIFT RETURN	141	┐	174
	10	+	43	L	76	╲	109	PASSAGGIO AI MAIUSCOLO	142	▂	175
	11	,	44	M	77	╱	110		143	┌	176
	12	–	45	N	78	┌	111	BLK	144	┴	177
RETURN	13	.	46	O	79	┐	112	CRSR ↑	145	┬	178
PASSAGGIO IN NEGATIVO	14	/	47	P	80	●	113	RVS OFF	146	┤	179
	15	0	48	Q	81	▁	114	CLR HOME	147	▎	180
	16	1	49	R	82	♥	115	INST DEL	148	▍	181
CRSR ↓	17	2	50	S	83	▏	116		149	▐	182
RVS ON	18	3	51	T	84	╭	117		150	▔	183
CLR HOME	19	4	52	U	85	╳	118		151	▀	184
INST DEL	20	5	53	V	86	○	119		152	▃	185
	21	6	54	W	87	♣	120		153	▕	186
	22	7	55	X	88	▕	121		154	▖	187
	23	8	56	Y	89	♦	122		155	▝	188
	24	9	57	Z	90	┼	123	PUR	156	┘	189
	25	:	58	[	91	▒	124	CRSR ←	157	▘	190
	26	;	59	£	92	│	125	YEL	158	▚	191
	27	<	60	]	93	π	126	CYN	159		
RED	28	=	61	↑	94	◥	127	SPACE	160		
CRSR →	29	>	62	←	95		128	▌	161		
GRN	30	?	63	─	96		129	▄	162		
BLU	31	@	64	♠	97		130	▔	163		
SPACE	32	A	65	│	98		131	▁	164		

CAR = CARATTERE
COD = CODICE

Codici	da 192 a 223	Identici ai codici	da 96 a 127
Codici	da 224 a 254	Identici ai codici	da 160 a 190
Codice	255	Identico al codice	126

Movimento con sprite farfalla CBM 64

```
100 rem *** movimento ***
120 poKe53280,9:poKe53281,9:print"█"
1000 v=53248
1020 forK=0to62:reada:poKe15744+K,a:nextK
1040 poKev+39,7
1060 poKe2040,246
1080 x=50:y=50
1100 poKev,x:poKev+1,y:poKev+21,1
1200 form=1to8
1300 fors=1to160
1400 onmgosub2000,2100,2200,2300,2400,2500,2600,2700
1420 poKev,x:poKev+1,y
1500 nexts
1600 ifm=6thenx=210:y=50
1700 nextm
1800 end
2000 x=x+1:return
2100 y=y+1:return
2200 x=x-1:return
2300 y=y-1:return
2400 x=x+1:y=y+1:return
2500 x=x-1:y=y-1:return
2600 x=x-1:y=y+1:return
2700 x=x+1:y=y-1:return
3000 rem *** sprite farfalla ***
3010 data2,0,64,49,0,140
3020 data120,129,30,252,66,63
3030 data254,36,127,255,24,255
3040 data255,153,255,255,219,255
3050 data255,255,255,255,255,255
3060 data255,255,255,255,255,255
3070 data255,255,255,255,255,255
3080 data255,255,255,255,255,255
3090 data255,219,255,127,153,254
3100 data63,24,252,30,24,120
3110 data12,24,48
```

Sprite lettera "G" CBM 64

```
3000 rem *** sprite lettera g ***
3010 data255
3020 data255,255,255,255,255
3030 data224,0,7,224,0
3040 data7,231,255,231,231
3050 data255,231,231,255,255
3060 data231,255,255,231,255
3070 data255,231,255,255,231
3080 data255,255,231,240,7
3090 data231,240,7,231,255
3100 data231,231,255,231,231
3110 data255,231,231,255,231
3120 data224,0,7,224,0
3130 data7,255,255,255,255
3140 data255,255
```

Sprite lettera "J" CBM 64

```
3000 rem *** lettera j ***
3010 data255,255
3020 data255,255,255,255,224
3030 data0,7,224,0,7
3040 data255,227,255,255,227
3050 data255,255,227,255,255
3060 data227,255,255,227,255
3070 data255,227,255,255,227
3080 data255,255,227,255,231
3090 data227,255,231,227,255
3100 data231,227,255,231,227
3110 data255,231,227,255,224
3120 data3,255,224,3,255
3130 data255,255,255,255,255
3140 data255
```

Sprite tazza caffé CBM 64

```
3000 rem *** sprite tazza caffe' ***
3010 data8
3020 data66,0,4,33,0
3030 data2,16,128,4,33
3040 data0,8,66,0,0
3050 data0,0,255,255,192
3060 data255,255,192,245,85
3070 data252,245,85,254,245
3080 data85,199,245,85,195
3090 data245,85,195,245,85
3100 data195,245,85,199,245
3110 data85,206,245,85,252
3120 data245,85,248,245,85
3130 data240,245,85,192,255
3140 data255,192
```

Sprite gabbia zoo CBM 64

```
3000 rem *** sprite gabbia zoo ***
3010 data255,255,255
3020 data255,255,255,132,16
3030 data65,132,16,65,132
3040 data16,65,132,16,65
3050 data132,16,65,132,16
3060 data65,132,16,65,132
3070 data16,65,132,16,65
3080 data132,16,65,132,16
3090 data65,132,16,65,132
3100 data16,65,132,16,65
3110 data132,16,65,132,16
3120 data65,132,16,65,255
3130 data255,255,255,255,255
```

Sprite coniglio CBM 64

```
3000 rem *** sprite coniglio ***
3010 data0
3020 data0,0,255,0,255
3030 data255,129,255,15,199
3040 data224,3,199,128,0
3050 data238,0,0,238,0
3060 data0,238,0,0,255
3070 data0,1,255,128,3
3080 data147,192,7,147,224
3090 data15,147,240,31,239
3100 data248,31,125,248,15
3110 data131,240,7,131,224
3120 data3,131,192,1,255
3130 data128,0,127,0,0
3140 data0,0
```

Sprite lupo CBM 64

```
3000 rem *** sprite lupo ***
3010 data0,0,0,0,0
3020 data0,0,0,48,1
3030 data192,112,0,240,240
3040 data0,121,224,0,61
3050 data192,0,31,192,0
3060 data15,224,0,63,224
3070 data0,255,240,127,199
3080 data248,255,255,252,255
3090 data255,254,255,255,254
3100 data128,255,254,42,255
3110 data254,127,255,252,1
3120 data255,248,1,255,240
3130 data1,255,240
```